

Tarea 1

Enunciado



Objetivo

El objetivo general de esta tarea es que el alumno se familiarice con un *framework* de desarrollo web y que comprenda los desafíos básicos de realizar un *deploy* en la nube para dejar disponible para el público un programa web.

Objetivos específicos

Se espera que después de esta tarea, el alumno sea capaz de:

1. Crear un sitio web simple.
2. Manejar bases de datos y un CRUD básico mediante el uso del *framework*.
3. Realizar un *deploy* en algún proveedor de *cloud* y dejar el sitio disponible para el público.

Trabajo a realizar

Se solicita realizar un sitio web básico de noticias con una administración sencilla que cumpla con las historias de usuario que se listan a continuación.

Historias de usuario

Ver todas las noticias

Como usuario, quiero entrar al sitio web y que en la página principal se muestre un listado de las últimas 10 noticias publicadas, ordenadas desde la más nueva a la más antigua.

Para cada noticia, quiero ver el titular, la bajada, que por ningún motivo debe superar los 200 caracteres, el cuerpo de la noticia y un link a la noticia completa. Cuando el cuerpo de la noticia supere los 500 caracteres, debe truncarse en la última palabra antes de los 500 caracteres y poner 3 puntos adyacentes a la última palabra (ejemplo: *última palabra...*) para señalar que el texto continúa.

Por último, quiero ver la cantidad de comentarios publicados en cada noticia.

Ver detalle de noticia

Como usuario, cuando hago click en una noticia, quiero que el sitio me muestre una noticia completa, con su título, bajada y cuerpo correspondiente.

Además, quiero ver los comentarios realizados en la noticia y la posibilidad de agregar un nuevo comentario.

Para publicar un comentario, debo ingresar mi nombre y el contenido del comentario, y ambos campos deben ser obligatorios.



Ver sitio de administración

Si yo soy un usuario administrador del sitio, quiero poder ingresar a un sitio donde pueda ver y administrar todas las noticias publicadas en el portal.

Para ingresar al sitio de administración, debo iniciar sesión con un usuario y contraseña previamente creados.

El ingreso al sitio de administración deberé realizarlo por medio de un botón o link que estará presente en el sitio principal.

Las noticias presentadas en este portal de administración se me deben mostrar como una lista ordenada de más nueva a más antigua, y cada entrada de la lista debe darme la opción de editar o borrar una noticia específica.

Al hacer click en editar, se debe mostrar una vista donde se pueda cambiar el título, la bajada o el cuerpo de la noticia. No se debe poder editar la fecha de publicación.

Al hacer click en eliminar, se debe mostrar un diálogo de confirmación, el cuál al aceptar, la noticia se debe eliminar y debe desaparecer del listado. Al eliminarse una noticia, también se debe dejar de mostrar en el sitio principal.

Publicar noticia

En el sitio de administración, debe haber un botón que me permita crear una nueva noticia. Cuando se presione, se me debe mostrar un formulario en el que pueda ingresar el título, la bajada y el cuerpo de la noticia. Además, al guardar, se debe registrar la fecha y hora de la publicación de la noticia.

Luego de publicar una noticia, esta debe aparecer en el primer lugar del sitio principal y además se debe mostrar en el listado del sitio de administración.

Requisitos no funcionales

Lenguaje de programación y framework

Se recomienda el uso de Ruby, versión 2.3.0 o superior (<http://www.ruby-lang.org/>), y el framework Ruby On Rails, versión 5.0 o superior (<http://rubyonrails.org/>).

Arquitectura de software

El software por construir deberá estar construido en un *framework* usando el patrón MVC.

Bases de datos

Las noticias deberán ser guardadas en una base de datos PostgreSQL. Todas las consultas a la base de datos (SELECT, INSERT, UPDATE y DELETE) deberán ser realizadas a



través del ORM del *framework*. No deben quedar consultas en duro o conexiones directas a las bases de datos.

Publicación del sitio

El sitio debe ser alojado en un servidor Heroku. La capa gratuita del servicio es suficiente para el alcance de este requerimiento.

Versionamiento del código

La totalidad del código deberá ser versionado en un repositorio Github y deberá quedar en el *branch master*.

Documentación

Se deberá entregar un modelo UML de base de datos donde se indique claramente las relaciones entre las tablas, los campos de cada tabla, sus claves primarias y claves foráneas.

Gestión de usuarios

Los permisos de usuarios y seguridad de las páginas con inicio de sesión se deben manejar con los elementos del *framework* seleccionado para este fin. Esto quiere decir que no se puede usar un login con usuario/contraseña en duro y que los usuarios y permisos deben quedar en base de datos.

Para el manejo de usuarios y sesiones en Rails, se recomienda el uso de la gema *devise*.



Evaluación

Requisitos mínimos

Las tareas que no cumplan las siguientes condiciones no serán corregidas y serán evaluados con la nota mínima:

- El sitio deberá estar publicado en un servidor Heroku
- El código deberá estar versionado en su totalidad en un repositorio Git
- El sitio debe reflejar fielmente el código entregado en el repositorio
 - Se recomienda configurar un deploy automático en Heroku asociado al repositorio Github

Fecha de entrega

La tarea se deberá entregar el día 19 de marzo antes de las 15:30 hrs.

Penalizaciones

Se descontará 0,5 puntos de la nota de la tarea por cada hora de atraso en la entrega, contados a partir de la fecha estipulada en el punto anterior.

Cualquier intento de copia, plagio o acto deshonesto en el desarrollo de la tarea, será penalizado con nota 1,1 de acuerdo a la política de integridad académica del DCC¹.

Entregables

Cada alumno deberá entregar, mediante un formulario publicado en el sitio del curso, los siguientes recursos:

- URL del sitio implementado
- URL del repositorio Github
- Documento modelo de base de datos
- Usuario y contraseña usuario para publicar

¹ <http://www.ing.uc.cl/ciencia-de-la-computacion/programas/licenciatura/politica-de-integridad-academica/>



Rúbrica de evaluación

Requisitos funcionales

Vista	Funcionalidad	Criterio de aceptación	Puntaje Máximo
Ver todas las noticias	Sitio inicial	Al entrar al sitio inicial, se ven máximo 10 noticias	2
	Orden noticias	Las noticias se ordenan desde la más reciente a la más antigua	1
	Estructura noticias	Cada noticia tiene título, bajada, cuerpo y fecha de publicación	1
	Estructura cuerpo noticia	El cuerpo de la noticia se corta si tiene más del máximo de caracteres establecidos	1
	Número de comentarios	Cada noticia muestra el número de comentarios asociados a esta	1
Ver detalle de noticia	Sitio de noticia	La noticia mostrada tiene título, bajada, cuerpo completo y fecha de publicación	2
	Ver comentarios	La noticia muestra todos los comentarios asociados a esta	1
	Mostrar formulario de comentarios	La página muestra un formulario solicitando nombre y texto, y ambos campos son obligatorios	1
	Publicar comentario	El formulario permite publicar un nuevo comentario	1
Ver sitio de administración	Página inicial	Se muestran todas las noticias que han sido publicadas	2
	Inicio de sesión	Se pide usuario y contraseña para ingresar al sitio	2
	Orden noticias	Las noticias se ordenan desde la más reciente a la más antigua	1
	Estructura noticias	Cada noticia tiene título, fecha de publicación y se presentan las opciones de editar y eliminar	1
	Edición de noticias	Se puede editar cualquier campo de una noticia (excepto fecha de ingreso) y este cambio se ve reflejado en el sitio principal	1
	Eliminar noticia	Al eliminar una noticia, se deja de ver en el sitio de administración y en el sitio principal	1
Publicar noticia	Botón publicar noticia	Se puede ingresar una nueva noticia desde el sitio de administración	1



	Formulario publicación	El formulario de publicación permite ingresar título, bajada y cuerpo de la noticia	1
	Publicación	Al crear una nueva noticia, se publica y se muestra en el portal de noticias y en el portal de administración	1
	Fecha publicación	Al publicar una noticia, se ingresa automáticamente la fecha, la que no puede ser editada por el usuario	1

Requisitos no funcionales

Requisito	Criterio de aceptación		Puntaje Máximo
Arquitectura de software	Correcto uso del <i>framework</i>	Revisión general sobre el buen uso del <i>framework</i> , utilizando principios y reglas comúnmente aceptadas	2
	Correcto uso de controladores	Los controladores son "delgados", es decir, tienen mínima lógica y sólo se dedican a conectar modelos y pasar datos a la vista	1
	Correcto uso de modelos	El modelo se encarga de todas las consultas de datos, así como la lógica de éstas Ejemplo: Para el home, desde el modelo se obtienen las noticias ya cortadas en 1000 caracteres	2
	Correcto uso de vistas	Las vistas sólo imprimen datos, no se preocupan de la lógica	1
	Gemas correctamente referenciadas	Se referencian las gemas utilizadas correctamente en la configuración del proyecto	1
Bases de datos	Uso del ORM	Se utiliza el ORM del <i>framework</i> para las consultas a bases de datos, y no se usan queries directas.	2
Versionamiento del código	Repositorio	El código está versionado en su totalidad en la rama master del repositorio	1
Documentación	Modelo de bases de datos	El modelo entregado muestra claramente las tablas utilizadas, las claves primarias y foráneas	3



Pontificia Universidad Católica de Chile
Escuela de Ingeniería
Departamento Ciencias de la Computación
IIC3103 – Taller de integración
Profesor: Arturo Tagle

Bonus

Se entregará un bonus de 0,5 puntos adicionales a la nota final a aquellos alumnos que entreguen la tarea aplicando un *framework* de diseño como Bootstrap, Angular Material, Foundation o similar.



Recursos adicionales

A continuación, se presentan una serie de links que le podrán ser útiles para el desarrollo de la tarea.

Cursos sobre Ruby y Ruby on Rails

Rubymonk

Curso para aprender Ruby desde cero. Incluye ejemplos y una consola interactiva para ir ensayando y realizando ejemplos en el mismo navegador.

Link: <https://rubymonk.com/>

Learn Ruby the Hard Way

Saltándote los ejercicios, este curso se puede completar en sólo 4 horas. El sitio también incluye cursos para otros lenguajes.

Link: <https://learnrubythehardway.org/book/>

Codecademy

Curso que consiste en crear 8 aplicaciones en Rails, comenzando con lo más básico (arquitectura MVC, bases de datos, etc). Algunas partes de este curso son pagadas.

Link: <https://www.codecademy.com/learn/learn-rails>

CodeSchool - Rails for Zombies

Sin duda, el curso más popular para aprender Rails. La versión gratuita del curso cubre todos los básicos y se completa en 2 a 3 horas según el sitio.

Link: <https://www.codeschool.com/courses/rails-for-zombies-redux>

Creando un proyecto en Rails

Getting started with Rails

Guía oficial del *framework* para inicializar un proyecto utilizando Rails. Incluye todos los pasos para crear un proyecto, inicializar la aplicación, crear un layout y un modelo básico utilizando el ORM del *framework*.

Clean Architecture in Ruby

Artículo que muestra como crear un proyecto en Rails usando una arquitectura MVC.

Link: <https://medium.com/@fbzga/clean-architecture-in-ruby-v2-d2293c78d4a6>



Pontificia Universidad Católica de Chile
Escuela de Ingeniería
Departamento Ciencias de la Computación
IIC3103 – Taller de integración
Profesor: Arturo Tagle

Resolviendo dudas

StackOverflow

Foro más popular para resolver dudas sobre código, *frameworks* y lenguajes de programación. Antes de preguntar, asegúrate que la pregunta no se ha realizado anteriormente.

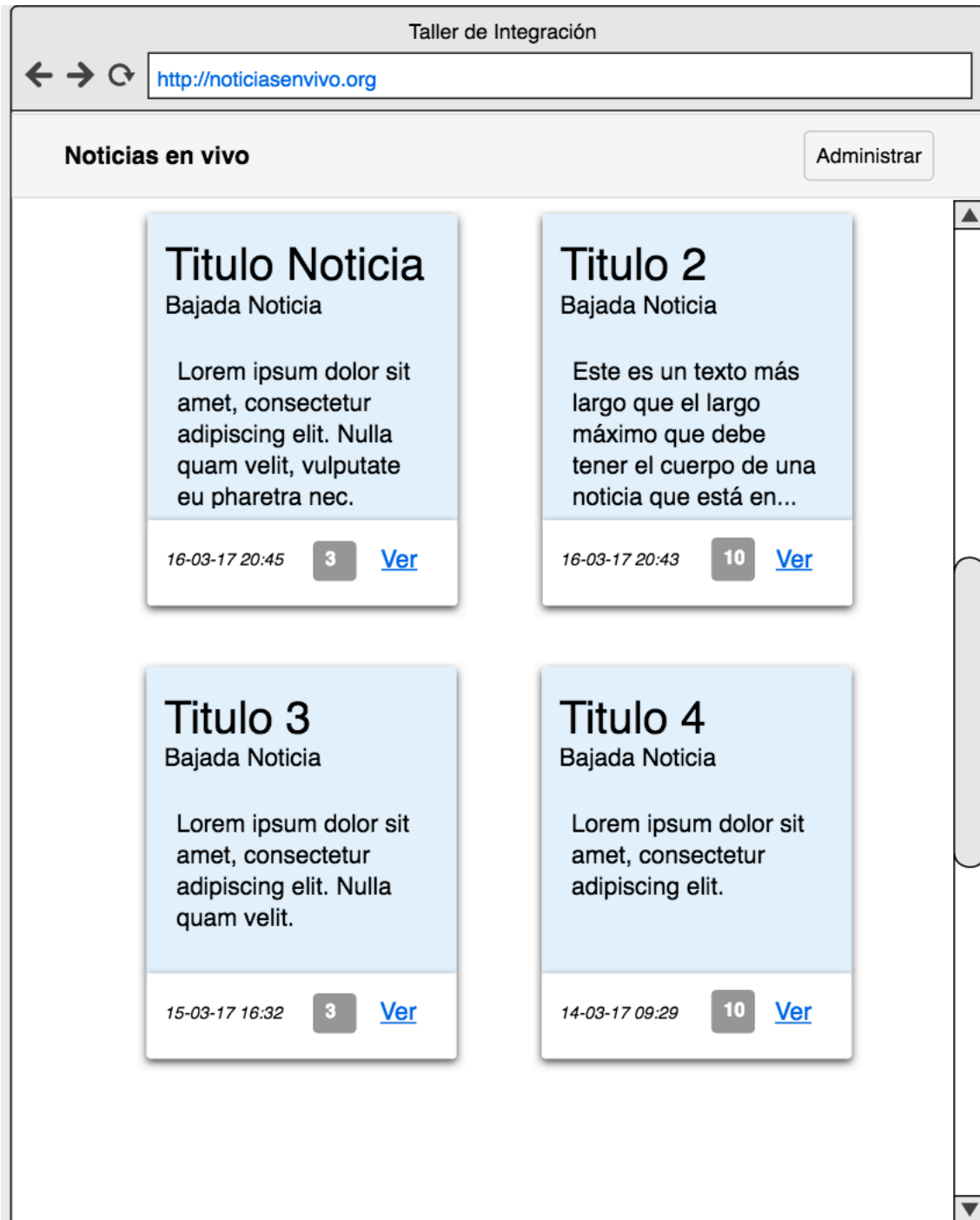
Link: <http://stackoverflow.com/>

Anexo 1

Ejemplos de página principal y página de administración



Ejemplo de página de portada





Ejemplo de página de administración

Taller de Integración

← → ↻

http://noticiasenvivo.org

Noticias en vivo

Ver Sitio

Administrar noticias

▼ Noticia	▼ Fecha Publicación	▼ Acciones
Titulo Noticia	16-03-17 20:45	Editar - Eliminar
Titulo 2	16-03-17 20:43	Editar - Eliminar
Titulo 3	15-03-17 16:32	Editar - Eliminar
Titulo 4	14-03-17 09:29	Editar - Eliminar

+

Publicar noticia